クロックレスARMプロセッサのアーキテクチャと非同期方式の効果

――非同期技術で低消費電力，低ピーク電流，低電磁放射を実現

デバイスの記事

渡辺信久

英国ARM社とオランダHandshake Solutions社は，ARM v5TE ISA準拠のクロックレス32ビット・プロセッサ・コア「ARM996HS」を共同開発した．これは，商用IPコアとして初めて非同期技術を用いて実装されたCPUコアである．非同期技術を用いることにより，同期方式の場合のほぼ1/3の消費電力で同等の性能を達成できた．また，回路内のピーク電流や外部に放出される電磁放射ノイズも低減された．　（編集部）

現在，ほとんどのディジタルLSIは，システムの全回路の動作を制御するグローバルな「クロック」を使用しています．複雑なシステムでは複数のクロックを使用することもあり，この場合は各クロック領域が同期サブシステムとして設計されています．

ご存じのように，クロック周波数は年々上昇しています．クロックの高速化に伴って，タイミング・スキューや消費電力の増大，電磁干渉の発生など，いくつかの問題がクローズアップされています．こうした問題を解決する手法の一つとして，非同期方式（クロックレス，またはセルフ・タイムドとも呼ぶ）のディジタル回路設計手法が注目を集めています．

ここでは，非同期方式を採用した初めての商用プロセッサIPコアである「ARM996HS」のアーキテクチャと，非同期化の効果について解説します．

1．同期方式の問題と非同期方式の利点

現在，主流となっている同期方式の設計には，以下の三つの問題があります．

第1の問題は，クロック周波数が上がるにつれて，グローバルな同期を維持するのが困難になることです．クロック・スキュー，すなわちチップ上の異なる点におけるクロック・タイミングのずれは性能を低下させ，最終的に回路の動作不良を招くこともあります．

第2の問題は，クロック周波数が高いと過度に電力を消費することです．クロックが高速になると，チップの総電力のかなりの部分をクロック分配ネットワークが消費します．クロック・ゲーティング技術（クロック配線にゲートを挿入し，クロックの供給・停止を制御して消費電力を低減する手法）はクロック・ネットのむだな動作をなくす効果はあるものの，その一方でクロック・スキューを悪化させるという欠点があります．

第3の問題は，チップ全体のグローバル同期が供給電流の過渡現象を最大化し，電磁干渉を引き起こすことです．EMC（electro magnetic compatibility；電磁環境適合性）規制は厳しくなりつつあり，高速なクロックを使いながらその規制に適合することが困難になりつつあります．

● 高速なクロックが引き起こす三つの問題を解決

非同期方式の設計は，上記の問題に対して以下のように対処します．

まず，非同期設計ではクロックがないので，そもそもクロック・スキューの問題は発生しません．また，非同期設計において回路に過渡現象が発生するのは，要求にこたえて必要な作業を実行するときだけです．つまり，余分な電

KeyWord　非同期方式，ARM996HS，クロックレス，ARMv5TE，Thumb，メモリ保護ユニット，マスク不可割り込み，ARM968E-S，ハンドシェイク・プロトコル，SAGE-X

力を消費しません（クロック信号によって過渡現象が継続的に発生することはない）．さらに，非同期回路では内部動作の干渉が少ないので，電磁放射ノイズも少なくなります（電磁環境適合性が向上する）．

クロック同期の回路は，プロセッサのクロック周波数，およびその高調波の電磁波を放射します．システムの中にRF（radio frequency）回路が存在する場合，そのようなスプリアス信号を本来のRF信号とまちがわないように配慮する必要があります．一方，クロックレスの非同期設計では，電磁放射が少ないだけでなく，比較的予測しやすい平たんな放射スペクトラム（ホワイト・ノイズ）を生成します．放射スペクトラムが平たんになり，さらにピーク電流が低くなれば，アナログ回路やRF回路とディジタル回路を統合しやすくなります．

● 低電力から高性能へ，モードを一瞬で切り替えられる

さらに非同期プロセッサには，以下のような利点もあります．

まず，ゼロ消費電力の動作モードと最大性能の動作モードの間を一瞬で切り替えることができます．組み込みアプリケーションの多くは，動作の途中で作業負荷が急激に変化します．そこで，割り込み待ちの間は動作を完全に停止してゼロ待機電力状態とし，割り込みが発生したときに瞬時に応答するようにします．こうすることで，プロセッサの消費電力を大幅に低減できる可能性があります．

また，クロックに依存しないので，モジュール化が容易になり，設計の自由度も上がります．同期方式では，クロック周波数が決まると，一般に使用できるコンポーネントの組み合わせが制限されます．

さらに，環境条件の変化に対する耐性が高いという利点もあります．つまり，温度や供給電源，電流の変化に追随しやすくなります．**図1**に示すように，非同期プロセッサの電圧・電流変化に対するマージンは，同期プロセッサと比べて格段に大きくなります．つまり，クロックレスの非同期回路は環境条件の変化に対する耐性に優れていると言えます．例えば今回のプロセッサ・コアは環境パラメータの変化を自動的に検出して適応するため，同期方式のコアの場合より幅広い環境条件で安定的に動作します．一般に，同期方式のコアが安定動作する範囲は，非同期方式のコアより大幅に狭くなります．

● なぜ非同期プロセッサは普及しなかったのか

以上のような多くの長所を持つにもかかわらず，なぜ，これまで非同期プロセッサはあまり普及してこなかったのでしょう．

その理由は，以下のようにいくつか考えられます．

- 非同期設計用の良い設計ツールがなかった（非同期設計に対するEDAツールの支援が手薄だった）
- 経験のある非同期回路の設計者が少なかった
- 同期方式中心の設計教育を通して，非同期技術への過度の不信感が植え付けられた

こうした理由により，非同期プロセッサが商用として開発されることは，これまでほとんどありませんでした．

2．非同期ARMプロセッサを開発

筆者ら（英国ARM社とオランダHandshake Solutions社，以下ARM社とHS社）は，本格的な商用化，IP製品化を目ざして，非同期プロセッサ・コア（ARM996HS）を共同開発しました．以下では，本プロセッサ・コアの概要について述べます．

本プロセッサ・コアは，専用のセル・ライブラリやトランジスタ・レベルの設計，カスタム・レイアウトなどを必要としない，ライセンス供給されたCPUコアです．既存の標準的な論理最適化，検証，レイアウトのフローに対応した柔軟なファームIP（ライブラリ依存のネットリスト）の形態で提供されます．本プロセッサ・コアは，オンチッ

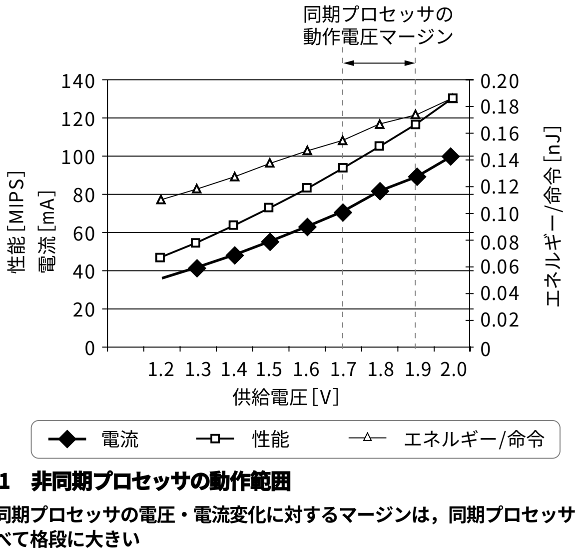


図1　非同期プロセッサの動作範囲

非同期プロセッサの電圧・電流変化に対するマージンは，同期プロセッサと比べて格段に大きい

プ・バスの標準となっているAMBA AHB（Advanced High-performance Bus）規格に対応しています．そのため，システムLSIに組み込む際のツールや手法がよく整備されており，比較的短時間のうちにシステムLSIに組み込むことができます．

先に述べたように，非同期方式を採用している本プロセッサには，低消費電力や低ピーク電流，低電磁放射ノイズなどの特徴があります．そこで，自動車や低コストの家電機器，ワイヤレス機器，医療用インプラント，スマート・カード，センサ・ネットワークなどのアプリケーションに利用されると筆者らは期待しています．

● セキュリティ機能やハードウェア除算器を搭載

本プロセッサ・コアは，ARM9Eファミリに属する新しい32ビットRISCプロセッサ・コアです．ARM9Eファミリは，16ビットのThumb命令セットと32ビットのARM命令セットを含むARMv5TE ISA（命令セット・アーキテクチャ）をベースとしています．

図2に本プロセッサ・コアのブロック図を示します．ハーバード・アーキテクチャを採用する本プロセッサ・コアは，5段の整数演算パイプラインと32ビット積和演算（MAC）ブロックから構成されています．また，命令用とデータ用の密結合メモリ（TCM）があり，それぞれの最大容量は4Mバイトです．命令インターフェースとデータ・インターフェースを提供するのは，AMBA AHB-Liteデュアル同期バスです．

また，本プロセッサ・コアは，セキュリティ拡張としてメモリ保護ユニット（MPU：memory protection unit）とマスク不可割り込み（NMI：non-maskable interrupt）の機能を備えています．さらに，ハードウェア除算コプロセッサも内蔵しています．

複数のソースから入手したソフトウェアを再利用するには，各モジュールを分離し，互いの干渉を最小限に抑えるメカニズムが必要です．すなわち，多くのソフトウェア・ルーチンを分離できなければなりません．これを実行するのがメモリ保護ユニットです．OSのカーネル内の各タスクを十分に隔離するため，本プロセッサ・コアは，各タスクに32バイトの細かいメモリ領域を割り当てます．これにより，メモリ・リソースを有効に使用し，複数のタスクに対応できます．このレベルの細かさがあれば，ユーザとシステム，タスクとタスク，データとデータ，スタックとスタックをきちんと分離できます．オーバラップした領域を使ってアクセスを共有することも可能です．

マスク不可割り込みは，ソフトウェアによるマスクを実行できなくした割り込みです．制御アプリケーションなど，プログラム実行の予測がとくに必要な場合，マスク不可割り込みの機能は必須です．マスク不可割り込みは，特定の時間に処理する必要のあるウォッチドッグ・タイマがシステム内に使用されている場合に非常に重要となります．

また，センサから読み出した値をスケーリングする必要のあるアプリケーションでは，ハードウェア除算命令が有効です．ハードウェア除算コプロセッサは，メインのプロセッサ・パイプラインと並行動作します．そして，通常のコプロセッサ命令を使って，符号なし，符号付きの32ビット除算を実行します．

● 5段の非同期パイプラインで命令を処理

本プロセッサ・コアは，内部でハンドシェイク・プロト

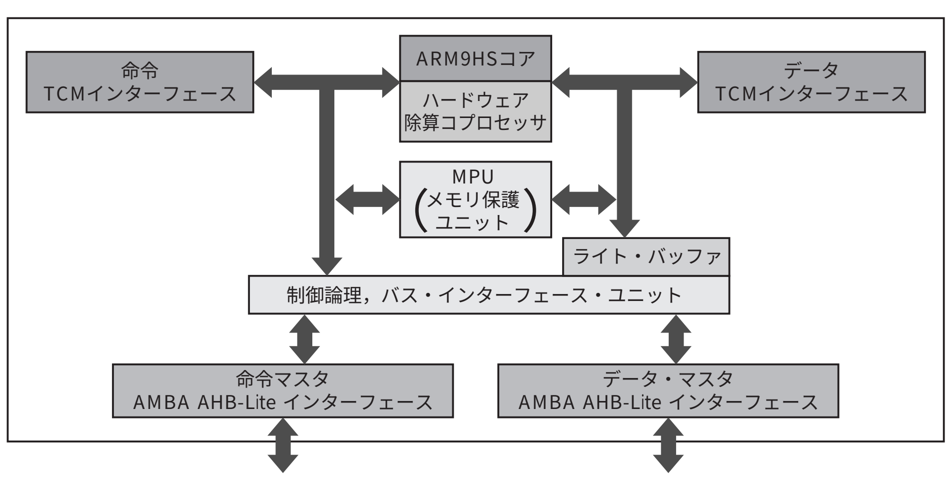


図2
ARM996HSプロセッサ・コアのブロック図
本プロセッサ・コアは，命令用とデータ用の密結合メモリ（TCM），メモリ保護ユニット，マスク不可割り込み，ハードウェア除算コプロセッサを備えている．AMBA AHB-Liteデュアル同期バスが命令インターフェースとデータ・インターフェースを提供する

コルを利用したモジュール間通信を行っています．CPUを構成する各モジュールはリクエスト（要求）信号とアクノリッジ（応答）信号を交わすハンドシェイクにより信号を交換します．**図3**に本プロセッサ・コアを構成する内部ブロックの基本回路を示します．

非同期プロセッサの設計では，それぞれのタイミング，それぞれの速度で動作しつつハンドシェイクする一連の非同期パイプラインを組み込むことができます．本プロセッサ・コアも，5段の非同期パイプラインを備えています（**図4**）．

同期プロセッサの設計では，パイプラインのサイクル速度を高めながら命令当たりの平均サイクル数を少なくするのが基本です．一方，非同期設計の「サイクル」は外部クロックによって決まるものではなく，時間は固定していません．

要求・応答のハンドシェイクを使って非同期パイプライン処理ユニットを構築する場合，各ステージに処理遅延を設け，ハンドシェイクによって結果を次のステージに送ります．同期パイプラインでは，全パイプラインがワースト・ケースの環境条件（電圧や温度など）におけるもっとも遅いステージの処理に合わせて動作するものとしてクロックを決めます．一方，非同期パイプラインは現在の条件によって決まる速度で動作します．まれに起こるワースト・ケースの条件によって，特定のユニットの処理に時間がかかることはあります．万一このような条件が発生したときには性能が低下することもありますが，その頻度が少ない

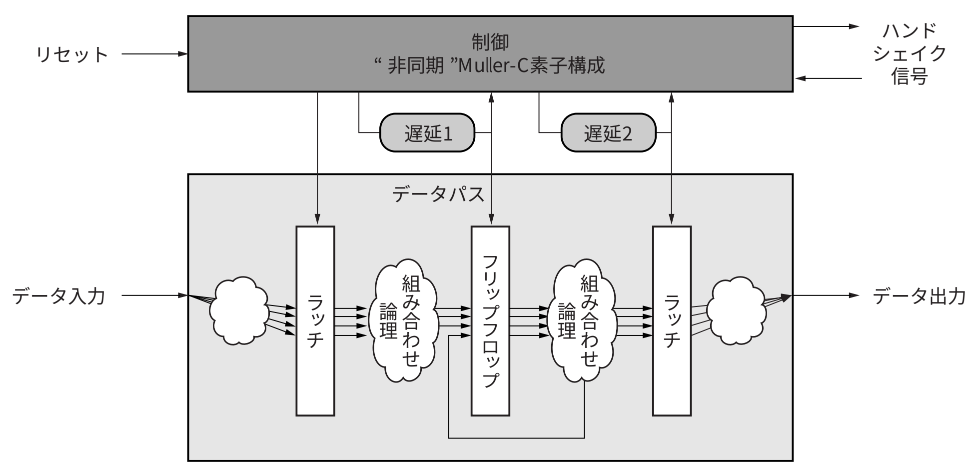


図3
内部ブロックの基本回路
本プロセッサ・コアは，内部でハンドシェイク・プロトコルを利用したモジュール間通信を行っている

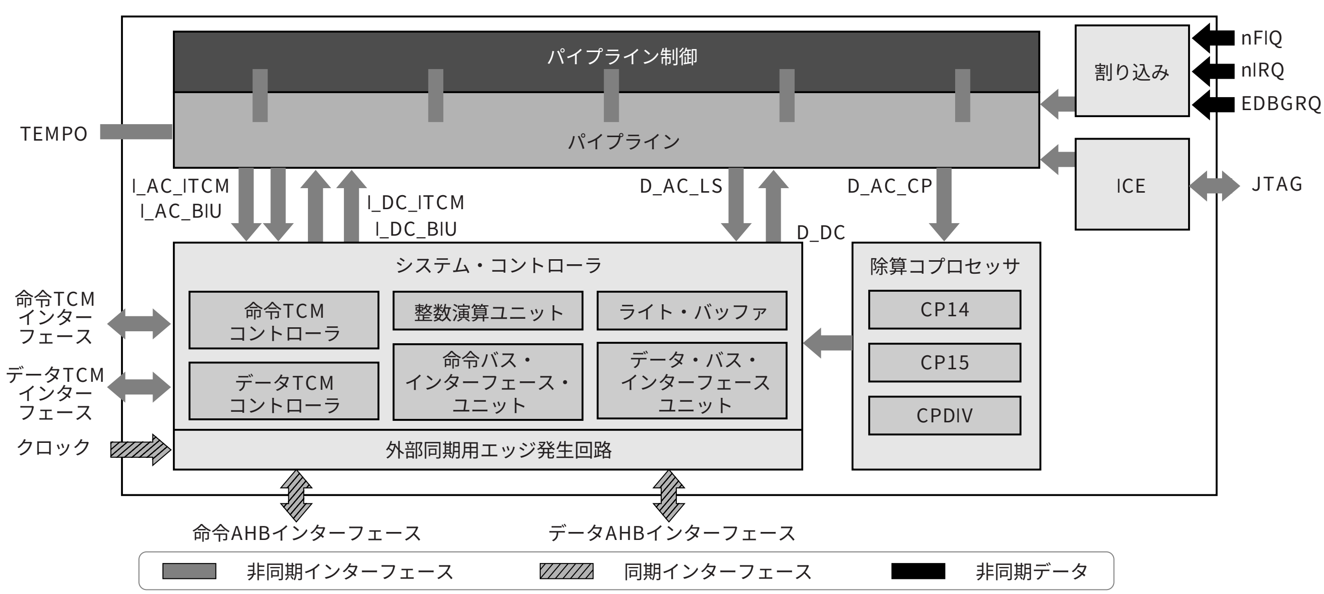


図4　演算処理部（ARM9HSコア）の構成
本プロセッサ・コアの演算処理部は，5段の非同期パイプラインを備えている

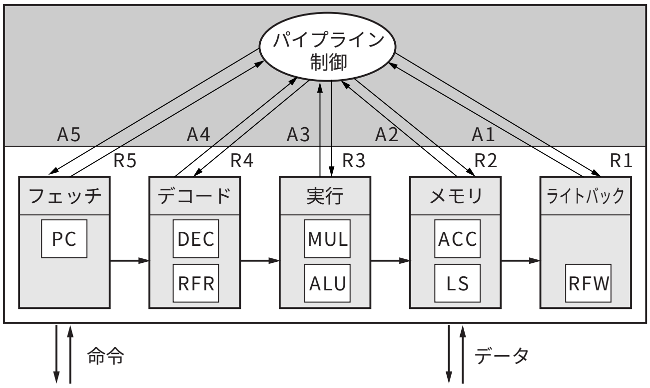


図5　非同期パイプライン
ARMv5TE ISAを実装した標準的なARM9E 5段パイプラインを採用している

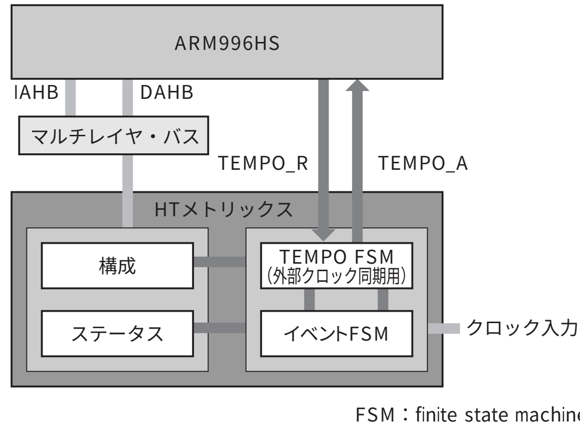


図6　HTメトリックス
HTメトリックス・ブロックはAHBバスに接続され，一つの周辺ブロック（ペリフェラル）として機能する

ので，性能全体に与える影響は小さいと考えられます．

本プロセッサ・コアは，ARMv5TE命令セット・アーキテクチャを実装した標準的なARM9E 5段パイプラインを採用しています．パイプラインの各ステージは，以下のとおりです（**図5**）．

- 命令プリフェッチ・ステージ
- デコード・レジスタ読み出しステージ
- 実行ステージ
- データ・メモリ・インターフェース・ステージ
- ライトバック（レジスタ書き込み）ステージ

同期パイプラインの場合と異なり，非同期パイプラインでは，パイプライン制御論理の処理はパイプライン中のデータ・エレメントのクロッキングと同時に実行されるのではなく，時間的に分散して実行されます．また，更新が必要なデータ・エレメントだけがクロッキングされます．さらに，パイプラインは命令フェッチやデータをロード・ストアするため，システム・コントローラとハンドシェイクによって通信します．

● 非同期回路の性能をコントロールするブロックを用意

本プロセッサ・コアには，デバッグやプロファイリングの支援用に「HT（Handshake Technology）メトリックス」と呼ぶ機能ブロックが用意されています．これにより，本プロセッサ・コアは命令を実行した後，例えば外部からのクロックを待ってパイプラインをクロックに同期させることができます．また，非同期回路の性能を固定クロック周期まで落とすことができるため，特別な環境を用意しなくてもワースト・ケースの動作を模擬できます．

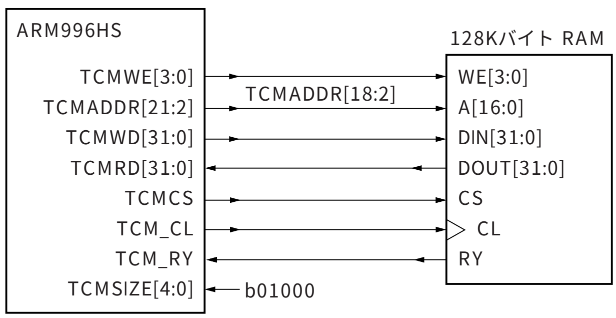
前述したように，ハンドシェイク回路はその特性により温度や供給電圧，供給電流といった環境の変化に自動的に適応できます．ただし，同期方式のコアの動作が破たんするような範囲でも正しく動作する強固さはありますが，回路性能が動作条件に依存するため，ワースト・ケースの環境条件を再現する（回路を低速にする）方法がありません．これを改善するために作られたのがHTメトリックス・ブロックです（例えば，本プロセッサ・コア用に書かれたソフトウェアがコーナ・ケースの電力・電圧・温度条件で正常に機能するかどうかを検証できる）．

HTメトリックス・ブロックはAHBに接続され，一つの周辺ブロック（ペリフェラル）として機能します（**図6**）．HTメトリックス・ブロックにより，ある程度の同期制御性と可観測性を非同期回路に付加することができます．

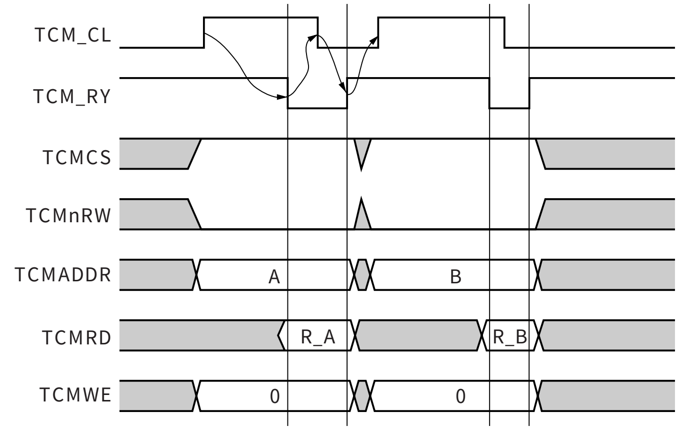
本プロセッサ・コアはシリコンが許容する限り高速に動作させることができますが，HTメトリックス・ブロックを使って人為的に性能を制限し，ファームウェアをあらゆる条件下で確実に動作させることもできます．こうした機能は，リアルタイム応用のソフトウェアを動作させる際に欠かせません．

HTメトリックス・ブロックは，以下のモードで実行するようにプログラムでき，実行時間中のモード切り替えも可能です．

- 各命令実行を外部イベント（例えばHTメトリックス・ブロックに与えられるクロック）に同期化させるモード
- N個の命令実行をM個の外部イベント（M個のクロック・パルスなど）に同期化させるモード
- 本プロセッサ・コアを，ブレーキをかけることなく最高



(a) バス構成



(b) リード/ライトのタイミング図

図7　密結合メモリ(TCM)のメモリ・アクセス
密結合メモリとして，通常の同期型RAMを集積できる．メモリ関連の周辺回路を用意する必要はない

表1　ARM996HSとARM968E-Sの仕様

	ARM996HS	ARM968E-S
ARM ISA(命令セット・アーキテクチャ)	ARMv5TE	ARMv5TE
コア論理	クロックレス	同期
パイプライン段数	5段	5段
Thumb-1 (16ビット命令)	対応	対応
DSP拡張	対応	対応
VFP9 FPU(浮動小数点演算ユニット)	なし	オプション
Java拡張	なし	なし
ハードウェア除算コプロセッサ	対応	なし
NMI (マスク不可割り込み)	対応	なし
キャッシュ・メモリ	なし	なし
命令TCM(密結合メモリ)/データTCM	最大4Mバイト/最大4Mバイト	最大1Mバイト/最大1Mバイト
メモリ管理	MPU (メモリ保護ユニット)	なし
メインI/Oバス	AHB-Lite×2	AHB-Lite×1
バス・アーキテクチャ	ハーバード	フォンノイマン
総バス幅	64ビット	32ビット
提供形態	ライブラリ依存のネットリスト(ファームIP)	汎用のネットリスト(ソフトIP)
提供開始時期	2006年	2004年

表2　ARM996HSとARM968E-Sの性能，消費電力，ゲート数

	ARM996HS	ARM968E-S
周波数 (等価MHz)	50(ワースト・ケース，1.08V，125℃)	100
	77(通常，1.2V，25℃)	
性　能 (Dhrystone MIPS)	54(ワースト・ケース，1.08V，125℃)	107
	83(通常，1.2V，25℃)	
消費電力 (mW/等価MHz)	0.045	0.13
ゲート数 (2入力NAND換算)	89,000	88,000

注：等価MHzは，ARM968E-Sが同じ性能を達成するために必要な周波数．この例では77MHzで動作させている．

性能で走らせるモード
- ある一定時間に実行される命令数をカウントする性能測定のモード

メモリ・インターフェースの例として，**図7**にCPU(ARM996HS)と密結合メモリ(TCM)の間のインターフェースを示します．密結合メモリとしては，通常の同期型RAMを集積できます．メモリ関連の周辺回路を用意する必要はありません．リード・アクセスはシンプルで，本プロセッサ・コアはハンドシェイク・プロトコルによってRAM固有の速度に自動的に適応します．RAMとのハンドシェイクにはRAM Clock(CL)ピンとReady(RY)ピンを使用します．

● 非同期ARM996HS vs. 同期ARM968E-S

非同期方式の効果を確認するため，本プロセッサ・コア，およびこれとほぼ同じ仕様のクロック同期方式のコア(ARM968E-S)について，ベンチマーク測定を行いました．**表1**にARM996HSとARM968E-Sの概要を示します．

シミュレーションには，台湾TSMC(Taiwan Semiconductor Manufacturing Co.)の0.13μm CMOSプロセス用ライブラリ「SAGE-X」を利用してレイアウトした後のネットリストを使いました．また，ARM968E-Sのネットリストは，クロック周波数を100MHzに設定して合成しました．

消費電力の測定基準を明確にするため，どちらの設計も同等の性能を達成する条件で動作させてシミュレーション

しました（ARM968E-Sは77MHzで動作）．その結果を**表2**に示します．非同期方式のコアは，同期方式の場合のほぼ1/3の消費電力で同等の性能を達成していることがわかります．

3．非同期プロセッサの設計フロー

2004年10月にARM社とHS社は，ARMコアの実装について共同開発することを発表しました．目的は低消費電力，低ノイズ，耐環境性が同時に要求される新しいアプリケーションを想定した非同期プロセッサの開発でした．本稿で説明しているARM996HSプロセッサ・コアは，その最初の成果です．

HS社は，オランダRoyal Philips Electronics社の独立事業体です．この技術を組み込んだ多数の量産用設計が完了しており，すでに1億個以上ものICが製造，テスト，販売されているそうです．

● 設計入力と論理合成以外は従来ツールを利用する

先にも触れましたが，複雑なシステムの非同期回路による実装は，これまで簡単ではありませんでした．それは，以下のような問題があったからです．

- 非同期方式の設計フローが確立されておらず，それを支援するツールが十分に提供されていなかった
- 特殊なクロックレスのスタンダード・セル・ライブラリを用意する必要があった
- 非同期回路（セルフ・タイムド回路）を実装しようとすると制御論理の規模が増えるため，面積上のオーバヘッドが生じる
- 非同期回路のテスト手法が未発達

HS社は非同期回路の実装に関するこうした課題に取り組み，多くの設計が非同期方式で実装できることを実証してきています．

HS社の設計フローでは，専用の非同期セルなどを用意する必要はありません．従来のスタンダード・セル・ライブラリを使用します（変換作業は必要になる．詳しくは後述）．テクノロジ・マッピングや論理最適化，タイミング解析，テストは，現在標準的に使われている（同期方式向けの）EDAツールとの互換性をもちます．

設計入力には「Haste」と呼ばれる言語を使用します．これは，非同期回路の仕様と設計を記述するために開発された専用言語です．HasteはHDLのビヘイビア記述やC言語と似ています．Haste記述は，HS社が開発した「HTコンパイラ」に入力されます．HTコンパイラがVerilog HDLのネットリストを生成した後は，業界標準の実装フローに従ってレイアウト設計を行い，GDS IIフォーマットのマスク・データを作成します．つまり，自社のシステムLSIにARM996HSプロセッサ・コアを統合したいと考えているライセンシは，設計フロー全体をHS社のものと置き換える必要はありません．Verilog HDLのネットリストをIPブロックとして組み込むことで実現できます．スキャン・テストとATPG（automatic test pattern generation）も従来の標準的な手法をそのまま利用できます．

非同期方式の回路設計には，「ハンドシェイク・プロトコルを利用するため，信号数が多くなる」，「面積的に非効率なデータ・エンコード（例えば，2相2線式）を採用することが多い」といった理由で，チップ面積が大きくなるという評価があります．この問題に対してHS社の手法では，バンドルド・データ方式と呼ぶ1線式のエンコードを採用しています．これにより，同期方式の場合と同じデータパス論理やスタンダード・セル・ライブラリを利用できます．ゲート数も同期方式とほぼ同等で，高いレイアウト密度を実現しています．

● スタンダード・セル・ライブラリの変換作業が必要

本プロセッサ・コアの設計に使われたフローを**図8**に示します．

ハンドシェイク回路は，ハンドシェイク・セル・ライブラリとして用意されます．まず，それぞれのハンドシェイク・セルを，顧客のスタンダード・セル・ライブラリを使った形に翻訳（変換）します．この翻訳作業はHS社によって行われ，設計者からは見えません．このような作業を行うためには，使用するスタンダード・セル・ライブラリの論理情報が必要となります．

HS社の設計技術は，同期方式のブロックと非同期方式のブロックの混載に対応しています．論理最適化やタイミング検証，ATPG，FPGAプロトタイピング，レイアウト設計については，標準的なサード・パーティ・ツールへのインターフェースが用意されています．

● 非同期回路の詳細は意識しなくてよい

非同期方式の専門家でなければ設計できないようでは，

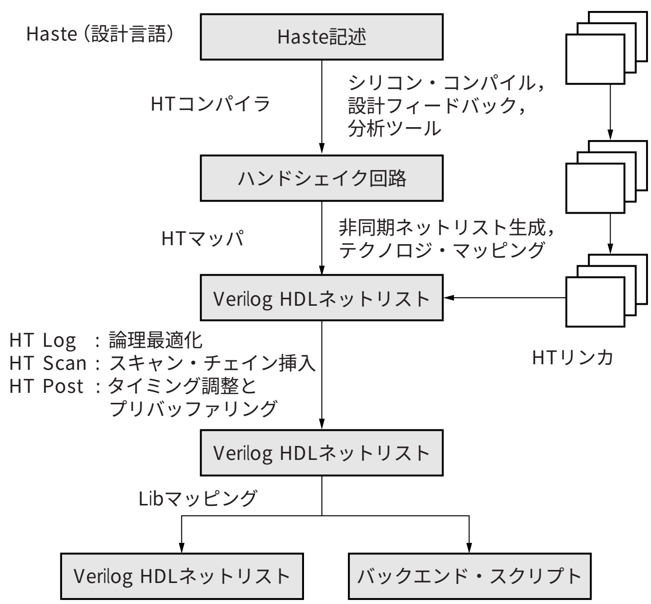


図8　本プロセッサ・コアの設計に使われたフロー
HTコンパイラがハンドシェイク回路を生成し，HTマッパがライブラリを使ってVerilog HDLネットリストを生成する

今回の手法は普及しません．そこで，回路を構造的にではなく，動作的に記述するHaste言語が開発されました．Haste言語を利用すると，非同期回路の詳細が隠ぺいされます．

ハンドシェイク・プロトコルの回路はHTコンパイラによって生成されるため，設計者がコードを細かく記述する必要はありません，そのため，非同期回路の落とし穴であるハザードやグリッチを避けることができます．

また，Hasteは，VHDLコードやVerilog HDLコードの再利用に対応しています．例えば，組み合わせ回路ブロックを`#include`文によって取り込めます．HTリンカは，ハンドシェイク用ラッパを組み合わせ回路ブロックの周りに挿入します．Hasteを利用すると，RTLのVHDL記述と比べて，20％程度記述ライン数が減ります．

● Verilog HDLネットリストなどが供給される

ARM996HSの設計データは，顧客のスタンダード・セル・ライブラリにマッピングされた（ファームIPの）ネットリストとしてHS社から提供されるか，またはSAGE-Xのような業界標準ライブラリを使ったネットリストとしてARM社から提供されます．また，このほかの設計データやタイミング制約ファイル，等価性チェッカ（フォーマル検証）用のデータ，ドキュメント，テスト・パターンなども ARM社で準備されます．

提供データには，上記のネットリスト（Verilog HDL）と配置配線を行うためのすべてのスクリプトが含まれています．特殊なメモリ・ブロックや特殊なセル・ライブラリ，カスタムの回路データ，特殊なプロセス技術を用意する必要はありません．

また，ARM996HSは従来のARM9Eファミリとデバッグ・アーキテクチャについて互換性があります．ソフトウェア開発用のコンパイラとデバッガは，ARM9E対応のものをそのまま利用できます．

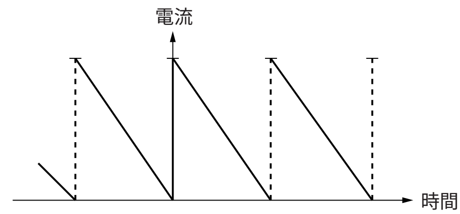


図9　電流の時間推移
同期方式の場合，クロック・エッジの直後で回路の活性度が最大となり，時間とともに低下していく

4．回路中のノイズと電磁放射の評価

ディジタル回路は，電源供給ラインへの電圧ノイズを生成します．そのため，同じ電源から電力の供給を受けているA-Dコンバータなどの動作を妨げる可能性があります．さらに，シリコン基板（サブストレート）へ電流を誘導し，同じシリコン基板上に集積されている回路の動作を妨げる可能性もあります．加えて，ディジタル回路では，そのクロック周波数の信号や高調波による電磁放射ノイズが発生し，RF回路を誤動作させる可能性があります．

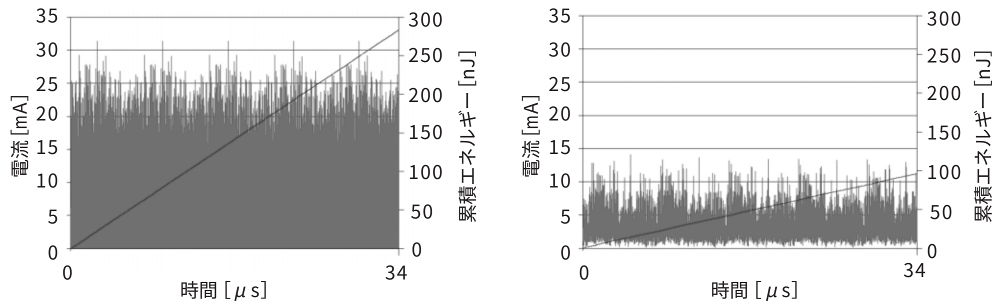
同期方式の場合，**図9**のようにクロック・エッジの直後で回路の活性度が最大となり，時間とともに低下していきます．この電流によってチップ上の電力消費とグラウンド・ラインのノイズが発生します．また，供給電圧の局所的な低下が起こり，性能と回路の信頼性に悪影響を与えます．

供給電流Iは下記の式で定義されます．

$$I = \frac{1}{2} - \frac{1}{\pi}\left(\sin\omega_c t + \frac{1}{2}\sin 2\omega_c t + \frac{1}{3}\sin 3\omega_c t + \cdots\right)$$
$$\omega_c = 2\pi f_c$$

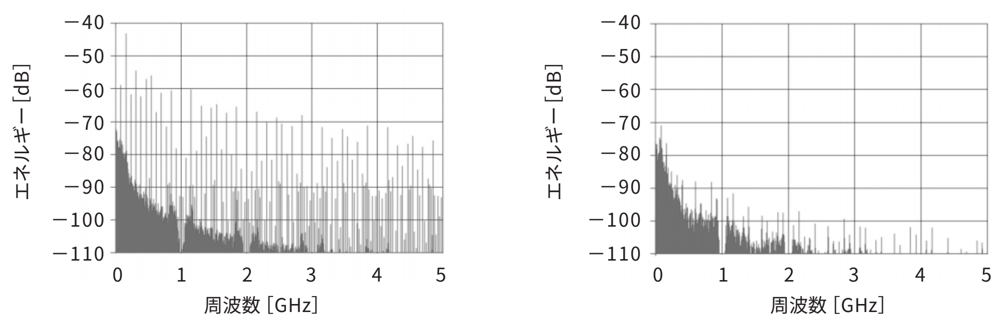
…………………………(1)

図10
電流と累積エネルギーの比較
(a)と比較して，(b)は低いピーク電流を実現していることがわかる．総電流量も半減している



(a) 同期方式 (ARM968E-S)　(b) 非同期方式 (ARM996HS)

図11
電磁放射ノイズの比較
(b)は全周波数(スペクトラム)にわたって電磁放射が少ない



(a) 同期方式 (ARM968E-S)　(b) 非同期方式 (ARM996HS)

このように，クロックは供給電流の周波数スペクトラムに直接影響します．

● 電流ピークは半分以下，電磁放射も大幅に低減

先ほどのベンチマーク測定と同じ条件で，非同期方式のプロセッサ・コア(ARM996HS)と同期方式のプロセッサ・コア(ARM966E-S)のピーク電流と電磁放射ノイズの比較を行いました．いずれのプロセッサ・コアも，TSMCの0.13μm CMOSプロセス用ライブラリ「SAGE-X」を使用して設計しています．

非同期方式では回路の動作が時間的に分散しているため，低ピーク電流となり，電力供給ラインのノイズを低減できます．**図10**は，電流と累積エネルギーのDhrystoneベンチマークの結果で，5回のテストの経過時間の値を示しています．非同期方式のプロセッサ・コアは，同期方式の場合よりも消費エネルギーが1/2.8(35.7％)少なく，電流ピークも1/2.4(41.7％)に減っています．

また，非同期方式では平たんな電磁スペクトラムになり，RF回路への電磁波妨害が減ります．**図11**はそれぞれのプロセッサ・コアから発生する信号の電磁スペクトラムと電流ピークの例です．すべてのスペクトラムについて，非同期方式のほうが電磁放射のピークが低減されています．

参考・引用*文献

(1) Arjan Bink, Richard York； ARM996HS Presentation in Spring Processor Forum‘06, 2006.
(2) ARM；ARM996HS Processor White Paper.
(3) Steve Furber； ARM system-on-chip architecture(Second edition), Addison-Wesley, 2000.

わたなべ のぶひさ
アーム(株) 取締役 エンジニアリングディレクター

<著者プロフィール>
渡辺信久．ロボット，からくり人形などの自動機械に興味を持つうちに，初期のマイクロコンピュータと出会う．以来，ゲート数削減のためシフト・レジスタをプログラム・カウンタに使った簡単な4ビット・マイコンや8ビット・マイコン，信号処理プロセッサの開発・設計を手がけ，オンチップ・マルチプロセッサの今日まで，プロセッサ技術の進展とともに歩んできた．今日のように分業化された設計フローでは経験することが難しくなったが，アーキテクチャ設計から論理，回路，セル・レイアウト設計，トランジスタの特性解析まで幅広く手がけることができたことは幸運だったと思っている．